

BBBBBBBBBBBB		AAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTT		LLL
BBBBBBBBBBBB		AAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTT		LLL
BBBBBBBBBBBB		AAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSS		RRRRRRRRRR		TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSS		RRRRRRRRRR		TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSS		RRRRRRRRRR		TTT		LLL
BBB	BBB	AAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA		SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA		SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA		SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA		SSS	RRR	RRR	TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLL

```
BBBBBBBB  AAAAAA  SSSSSSSS  LL  EEEEEEEEE  FFFFFFFF  TTTTTTTTTT
BBBBBBBB  AAAAAA  SSSSSSSS  LL  EEEEEEEEE  FFFFFFFF  TTTTTTTTTT
BB        BB  AA      AA  SS      LL  EE        FF        TT
BB        BB  AA      AA  SS      LL  EE        FF        TT
BB        BB  AA      AA  SS      LL  EE        FF        TT
BB        BB  AA      AA  SS      LL  EE        FF        TT
BBBBBBBB  AA      AA  SSSSSS  LL  EEEEEEE  FFFFFFFF  TT
BBBBBBBB  AA      AA  SSSSSS  LL  EEEEEEE  FFFFFFFF  TT
BB        BB  AAAAAAAAAA  SS      LL  EE        FF        TT
BB        BB  AAAAAAAAAA  SS      LL  EE        FF        TT
BB        BB  AA      AA  SS      LL  EE        FF        TT
BB        BB  AA      AA  SS      LL  EE        FF        TT
BBBBBBBB  AA      AA  SSSSSSSS  LLLLLLLLLL  EEEEEEEEE  FF
BBBBBBBB  AA      AA  SSSSSSSS  LLLLLLLLLL  EEEEEEEEE  FF
```

....
....
....
....

```
LL        IIIII  SSSSSSSS
LL        IIIII  SSSSSSSS
LL        II     SS
LL        II     SS
LL        II     SS
LL        II     SS
LL        II     SSSSSS
LL        II     SSSSSS
LL        II     SS
LL        II     SS
LL        II     SS
LL        II     SS
LLLLLLLLLL  IIIII  SSSSSSSS
LLLLLLLLLL  IIIII  SSSSSSSS
```

```
1 0001 0 MODULE BASSLEFT (
2 0002 0 IDENT = '1-005'
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *****
27 0027 1
28 0028 1
29 0029 1
30 0030 1 **
31 0031 1 FACILITY: String support library
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module extracts a substring according to the
36 0036 1 BASIC-PLUS-2 syntax. It finds the substring of a main string
37 0037 1 starting at the left end of the string (character position 1)
38 0038 1 and continues through the nth character of the string. This
39 0039 1 substring is copied to the destination string.
40 0040 1
41 0041 1 ENVIRONMENT: User mode, AST level or not or mixed
42 0042 1
43 0043 1 AUTHOR: R. Will, CREATION DATE: 19-Feb-79
44 0044 1
45 0045 1 MODIFIED BY:
46 0046 1
47 0047 1 R. Will, 19-Feb-79: VERSION 01
48 0048 1 01 - original
49 0049 1 02 - change linkage and call to COPY routine. 13-Mar-79 RW
50 0050 1 1-003 - Change string linkages to start with STR$. JBS 04-JUN-1979
51 0051 1 1-004 - Change call to STR$COPY. JBS 16-JUL-1979
52 0052 1 1-005 - CALL STR$LEFT. DELETE THIS MODULE WHEN COMPILER CHANGES. RW 31-OCT-79
53 0053 1
54 0054 1
55 0055 1 !<BLF/PAGE>
```


[illegible]

BASSLEFT
1-005

L 12
16-Sep-1984 00:41:59
14-Sep-1984 11:55:13

VAX-11 Bliss-32 V4.0-742
[BASRTL.SRC]BASLEFT.B32;1

Page 3
(3)

```

101 0193 1 GLOBAL ROUTINE BASSLEFT (DEST_DESC,      ! Pointer to destination descriptor
102 0194 1 SRC_DESC,                                ! Pointer to source descriptor
103 0195 1 CHAR_POS) : NOVALUE =                      ! Last character to be included
104 0196 1
105 0197 1
106 0198 1 ++
107 0199 1 FUNCTIONAL DESCRIPTION:
108 0200 1 This routine extracts the characters starting at the leftmost
109 0201 1 character (character position 1) and continuing through the
110 0202 1 character position specified by the input and copies that substring
111 0203 1 to the destination string (by JSB to STR$$COPY_R_R8) according
112 0204 1 to the syntax of the class of the destination string.
113 0205 1 If the input character position is > the length of the input string,
114 0206 1 then the length of the input string is used. If the input character
115 0207 1 position is < 1, the destination becomes a null string.
116 0208 1
117 0209 1 FORMAL PARAMETERS:
118 0210 1
119 0211 1 DEST_DESC.wt.dx      pointer to destination string descriptor
120 0212 1 SRC_DESC.rt.dx     pointer to source string descriptor
121 0213 1 CHAR_POS.rw.v      value of last character position to include
122 0214 1
123 0215 1 IMPLICIT INPUTS:
124 0216 1
125 0217 1 NONE
126 0218 1
127 0219 1 IMPLICIT OUTPUTS:
128 0220 1
129 0221 1 NONE
130 0222 1
131 0223 1 ROUTINE VALUE:
132 0224 1 COMPLETION CODES:
133 0225 1
134 0226 1 NONE
135 0227 1
136 0228 1 SIDE EFFECTS:
137 0229 1
138 0230 1 NONE
139 0231 1
140 0232 1 --
141 0233 1
142 0234 2 BEGIN
143 0235 2
144 0236 2 MAP
145 0237 2 DEST_DESC : REF BLOCK [8,BYTE];
146 0238 2
147 0239 2 MAP
148 0240 2 SRC_DESC : REF BLOCK [8,BYTE];
149 0241 2
150 0242 2 STR$LEFT (DEST_DESC [0,0,0,0], SRC_DESC [0,0,0,0], CHAR_POS);
151 0243 2 RETURN;
152 0244 1 END;                                !End of BASSLEFT
```

.TITLE BASSLEFT
.IDENT \1-005\

BASSLEFT
1-005

M 12
16-Sep-1984 00:41:59 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 11:55:13 [BASRTL.SRC]BASLEFT.B32;1

Page 4
(3)

.EXTRN STR\$LEFT

.PSECT _BAS\$CODE,NOWRT, SHR, PIC,2

.ENTRY BASSLEFT, Save nothing

PUSHAB CHAR_POS

MOVQ DEST_DESC, -(SP)

CALLS #3, STR\$LEFT

RET

: 0193

: 0242

: 0244

00000000G 7E 00
0C AC 0000 0000
04 AC 9F 00002
03 FB 00005
04 00010

: Routine Size: 17 bytes, Routine Base: _BAS\$CODE + 0000

: 153 0245 1
: 154 0246 1 END
: 155 0247 1
: 156 0248 0 ELUDOM

!End of module

PSECT SUMMARY

Name	Bytes	Attributes
_BAS\$CODE	17	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:BASLEFT/OBJ=OBJ\$:BASLEFT MSRC\$:BASLEFT/UPDATE=(ENH\$:BASLEFT)

: Size: 17 code + 0 data bytes
: Run Time: 00:01.7
: Elapsed Time: 00:03.5
: Lines/CPU Min: 8752
: Lexemes/CPU-Min: 23294
: Memory Used: 18 pages
: Compilation Complete

0024 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASINIGSC
LIS

BASINIT
LIS

BASINIDEF
LIS

BASINIDFS
LIS

BASINIGSB
LIS

BASINSTR
LIS

BASINTONE
LIS

BASLEFT
LIS

BASMARGIN
LIS

BASINTOL
LIS

BASKILL
LIS

BASTOBEG
LIS

BASTOEND
LIS

BASMATADD
LIS

BASMGAP
LIS